
interface.dtd

rules for describing an object interface in xml

```
<!ELEMENT   parameter   #PCDATA>
<!ATTLIST   parameter
            type           CDATA          #REQUIRED
            >

<!ELEMENT   method       (#PCDATA, (parameter)+)>
<!ELEMENT   object       (#PCDATA, (method)+)>
```

interface.h

example object interface in c

```
/* object */

typedef struct Stream {
    int id;
};

/* methods */

void StreamPlay (int id, int speed);
void StreamStop (int id);
```

interface.xml

the same object in xml using rules of interface.dtd

```
<object>Stream
  <method>Play
    <parameter type="int">id</parameter>
    <parameter type="int">speed</parameter>
  </method>
  <method>Stop
    <parameter type="int">id</parameter>
  </method>
</object>
```



call.dtd

rules for describing a c function call in xml

```
<!ELEMENT   parameter   #PCDATA>
<!ATTLIST   parameter
            value          CDATA          #REQUIRED
            >

<!ELEMENT   method      (#PCDATA, (parameter)+)>
<!ELEMENT   object       (#PCDATA)>
<!ELEMENT   call         (object,method)>
```

controller.c

example controller command in c

```
.
.
StreamPlay (0x01ae, 500);
.
.
```

call.xml

the same command in xml using call.dtd

```
<!-- example to play a stream -->
```

```
<call>
```

```
  <object>Stream</object>
```

```
  <method>Play</method>
```

```
  <parameter value="01ae">id</parameter>
```

```
  <parameter value="500">speed</parameter>
```

```
</call>
```

```

<!-------
      call.dtd

      rules for describing a c function call in xml
----->

```

```

<!--ELEMENT    parameter    #PCDATA>
<!--ATTLIST    parameter
               value          CDATA          #REQUIRED
               >

<!--ELEMENT    method      (#PCDATA, (parameter)+)>
<!--ELEMENT    object      (#PCDATA)>
<!--ELEMENT    call        (object, method)>

```

```

<!-------
      interface.dtd

      rules for describing an object interface in xml
----->

```

```

<!--ELEMENT    parameter    #PCDATA>
<!--ATTLIST    parameter
               type          CDATA          #REQUIRED
               >

<!--ELEMENT    method      (#PCDATA, (parameter)+)>
<!--ELEMENT    object      (#PCDATA, (method)+)>

```

```

<!-------
      interface.xml

      this document describes various CE services offered -a
      subset of the whole CE space.
----->

```

```

<?xml version="1.0"?>
<!DOCTYPE interface SYSTEM "interface.dtd">

<object>Stream

```

```

        <method>Open
            <parameter type="int">id</parameter>
            <parameter type="int">channel</parameter>
        </method>
        <method>Close
            <parameter type="int">id</parameter>
        </method>
    </object>

    <object>Control
        <method>Set
            <parameter type="int">id</parameter>
            <parameter type="int">level</parameter>
        </method>
    </object>

    <object>Storage
        <method>Open
            <parameter type="int">id</parameter>
            <parameter type="int">channel</parameter>
        </method>
        <method>Record
            <parameter type="int">id</parameter>
        </method>
        <method>Play
            <parameter type="int">id</parameter>
            <parameter type="int">speed</parameter>
        </method>
        <method>Stop
            <parameter type="int">id</parameter>
        </method>
        <method>Close
            <parameter type="int">id</parameter>
        </method>
    </object>

    <object>Display
        <method>Open
            <parameter type="int">id</parameter>
            <parameter type="int">channel</parameter>
        </method>
        <method>Render
            <parameter type="int">id</parameter>
        </method>
        <method>Blank

```

```

        <parameter type="int">id</parameter>
    </method>
    <method>Control
        <parameter type="int">id</parameter>
        <parameter type="int">cid</parameter>
        <parameter type="int">level</parameter>
    </method>
    <method>Close
        <parameter type="int">id</parameter>
    </method>
</object>

```

```

<!-------
    otr.xml

    an xml representation of one touch record

    c representation:

    StreamOpen (100, 2); /* play a stream (pushed by satellite feed) */
    StorageOpen (24, 2); /* open a storage service */
    StorageRecord (24); /* record the stream */
----->

```

```

<?xml version="1.0"?>
<!DOCTYPE interface SYSTEM "call.dtd">
<call>
    <object>Stream</object>
        <method>Open</method>
            <parameter value="100">id</parameter>
            <parameter value="2">channel</parameter>
    </call>
<call>
    <object>Storage</object>
        <method>Open</method>
            <parameter value="100">id</parameter>
            <parameter value="2">channel</parameter>
    </call>
<call>
    <object>Storage</object>
        <method>Record</method>
            <parameter value="100">id</parameter>
    </call>

```

consumer (document_file, doc)

+---document_file<server_home.dtd, server_auto.dtd>

+---doc (ave_commands, cal_commands, services_home, server_auto, server_samsung_web_site,
server_auto_ford_explorer_98,,)

+---ave_commands<...command_string->

+---cal_commands<...command_string->

+---services_home (client, av, lighting, comms, hvac, utility, security, appliance, convenience,,)

+---xml_utility (download_DTD_file,,)

+---client (acknowledge, attention, error, post_message, sound, stop_schedule, stop_all,,)

+---sound (alarm, ring, buzz,,)

+---server_av (source, sink)

+---source (service_id, media, rate, protocol, stream_format, controls_gen, controls_av,,)

+---sink (service_id, media, rate, protocol, stream_format, controls,,)

+---service_id (url,,)

+---media (tpt_stream, ram, disk, tape,,)

+---disk (name, number,,)

+---rate<value>

+---protocol (61883/1394, UDP/IP/Ethernet,,)

+---61883/1394 (isoch_ch_no)

+---stream_format (video, audio,,)

+---video (dv, mpeg2tpt, ddstpt, mpeg2psx, mpeg1080i-tpt,)

+---audio (mpeg3, ao-3, midi,,)

+---controls_gen (ping, process_info, setup,,)

+---controls_av (flow_control, tune, timer_record, ui_control,,)

+---process_info (s/w_id, h/w_id)

+---h/w_id (ser_no, manuf, model, class,,)

+---s/w_id (ser_no, exe_name, version,,)

+---setup (clock,,)

+---clock (hours, minutes, seconds)

+---time_record (tune, flow_control)

+---flow_control (play, stop, goto, record,,)

+---play (time_params)

+---record (time_params)

+---tune (send_epg, channel,,)

+---channel (number, id, time_params,,)

+---time_params (now, start, duration, end,,)

+---ui_control (display, acoustic)

+---display (brightness, contrast, color/tint, horiz_size, vert_size,,)

+---acoustic (volume, bass, treble, balance, fade,,)

APPENDIX 3

- +—lighting (sensors, lights, send_epg)
- +—sensors (living_room, sky,,)
- +—lights (rooms_up, rooms_down, yard,,)
 - +—rooms_up (bed1, bed2, bed3, bed4,,)
 - +—rooms_down (family, kitchen, living, dining, soho, garage,,)
 - +—yard (front, back)
 - +—bed1 (lamp, dimmer,,)
 - +—dimmer<value>
- +—comms (netman, intercom, telco,)
 - +—netman(send_device_list, send_configuration, send_smp_mib,,)
 - +—intercom()
 - +—telco()
- +—hvac (controls_gen, controls_hvac,,)
 - +—controls_hvac (a/c, heat, temp, humidity,)
 - +—temp (low, high, hysteresis,,)
- +—utility(meters, energy_mgmt,,)
 - +—meters (water, gas, electric,,)
 - +—water<value>, gas<value>, electric<value>
- +—security(sensors, send_epg, alarm,,)
 - +—sensors (peripheral, motion,,)
 - +—peripheral (rooms_up, rooms_down,,)
 - +—motion (rooms_down, yard,,)
- +—appliances (microwave, range, oven, fridge, freezer, coffee, toaster, washer, dryer, water_heater,,)
 - +—microwave (send_epg, controls,,)
 - +—fridge (temp,,)
 - +—water_heater (temp)
- +—convenience (window, curtain_open, door/gate, pool/spa, bath, fountain, lift,,)
 - +—curtain_open<value>
- +—server_auto (message, mileage, maintenance,,)
 - +—mileage <data>
 - +—maintenance <data>

APPENDIX 3, CONT.D

existing implementation:

```
void DeviceCALCommand (int command) {  
    .  
    .  
    /*  
    create CAL formatted byte string to represent this object/method  
    and output to the wire  
    */  
    CreateCALFormattedByteString(command); /* different for every  
                                           protocol */  
    SendCALByteString(); /* different for every protocol */  
}
```

wrapping the XML Service API call:

```
void DeviceCALCommand (int command) {  
    {  
        /*  
        replace CAL implementation with calls to the XML ServiceAPI  
        */  
        CreateXMLMessage (command); /* always the same */  
  
        SendXMLMessage (); /* always the same */  
    }  
}
```

APPENDIX 4


```

consumer (document_file, doc)

+----document_file<server_home.dtd, server_auto.dtd>

+----doc (services_home, server_auto, server_samsung_web_site,, avc_commands, cal_commands,,)

+----services_home (xml_utility, client, server_av, lighting, comms, hvac, utility, security,
appliances, convenience,,)

+----xml_utility (download_DTD_file,,)

+----client (acknowledge, attention, error, post_message, sound, stop_schedule, stop_all,,)

+----sound (alarm, ring, buzz,,)

+----server_av (controls_gen, source, sink)

+----controls_gen (ping, process_info, setup,,)

+----process_info (s/w_id, h/w_id)

+----h/w_id (ser_no, manuf, model, class,,)

+----s/w_id (ser_no, exe_name, version,,)

+----setup (clock,,)

+----clock (hours, minutes, seconds)

+----source (service_id, media, rate, protocol, stream_format, controls_av,,)

+----sink (service_id, media, rate, protocol, stream_format, controls_av,,)

+----service_id (url,,)

+----media (tpt_stream, ram, disk, tape,,)

+---- disk (name, number,,)

+----rate<value>

+----protocol (61883/1394, UDP/IP/Ethernet,,)

+----61883/1394 (isoch_ch_no)

+----stream_format (video, audio,,)

+----video (dv, mpeg2tpt, dsstpt, mpeg2pes, mpeg1080i-tpt,)

+----audio (mpeg3, ac-3, midi,,)

+----controls_av (flow_control, tune, timer_record, ui_control,,)

+----timer_record (tune, flow_control)

+----flow_control (play, stop, goto, record,,)
+----play (time_params)
+----record (time_params)
+----time_params (now, start, duration, end,,)

```

Appendix 5 (1 of 3)

```

+----tune (send_epg, channel,,)
+----channel (number, id, time_params,,)
+----ui_control (display, acoustic)
+----display (brightness, contrast, color/tint, horiz_size, vert_size,,)
+----acoustic (volume, bass, treble, balance, fade,)

+----lighting (sensors, lights, send_epg)

+----sensors (living_room, sky,,)

+----lights (rooms_up, rooms_down, yard,,)

+----rooms_up (bed1, bed2, bed3, bed4,,)

+----bed1 (lamp, dimmer,,)
+----dimmer<value>

+----rooms_down (family, kitchen, living, dining, soho, garage,,)

+----yard (front, back)

+----comms (homehub, intercom, telco,)

+----homehub(send_device_list, send_configuration, send_snmp_mib,,)

+----intercom()

+----telco()

+----hvac (controls_gen, controls_hvac,,)

+----controls_hvac (a/c, heat, temp, humidity,)

+----temp (low, high, hysteresis,,)

+----utility(meters, energy_mgmt,,)

+----meters (water, gas, electric,,)

+----water<value>, gas<value>, electric<value>

+----security(sensors, send_epg, alarm,,)

+----sensors (peripheral, motion,,)

+----peripheral (rooms_up, rooms_down,,)

+----motion (rooms_down, yard,,)

+----appliances (microwave, range, oven, fridge, freezer, coffee, toaster, washer, dryer,
water_heater,,)

+----microwave (send_epg, controls,,)

+----fridge (temp,,)

+----water_heater (temp)

+----convenience (window, curtain_open, door/gate, pool/spa, bath, fountain, lift, jacuzzi,,)

+----curtain_open<value>

+----server_auto (message, server_auto_ford_explorer_98,,)

+---- server_auto_ford_explorer_98 (mileage, maintenance,, )

```

Appendix 5 (2 of 3)

```
+----mileage <data>

+---- maintenance <data>

+----server_samsung_web_site(message, service, help,,)

+----avc_commands<...command_string...>
      +----service_id (url,,)

+----cal_commands<...command_string...>
      +----service_id (url,,)
```

Appendix S (3 of 3)